Software Development Methodologies

Lecturer: **Raman Ramsin**

Lecture 14

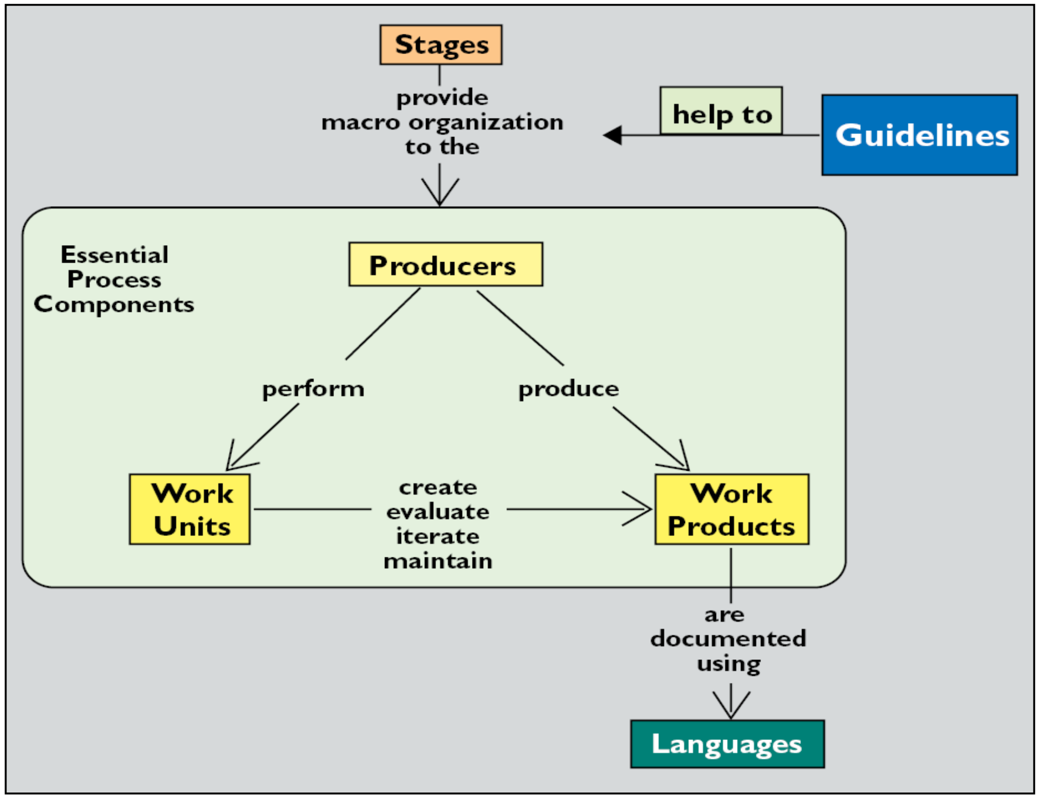
Process Metamodels

Process Metamodels

- Results of applying abstraction to software development processes
- Highlight the high-level features of a process or family of processes
- Can be instantiated in order to produce concrete processes
- The two most well-known object-oriented process metamodels:
  - OPEN Consortium's *OPEN Process Framework (OPF)*
  - OMG's *Software and Systems Process Engineering Metamodel (SPEM 2.0)*

OPEN Process Framework (OPF)

- A process metamodel defining five classes of components and guidelines for constructing customized OPEN processes
- Complemented by a component library from which process-component instances can be selected and assembled to create a specific process



For each element (represented by box), OPEN permits the user to select how many and which instances will be used. The OPF documentation provides a comprehensive list of suggestions on the best selections together with guidelines on their best organization.

[Firesmith and Henderson-Sellers 2001]

OPF: Component Classes

- *Work Products:* any significant thing of value (document, diagram, model, class, application) developed during the project.
- *Languages:* the media used to document work products, such as natural languages, modeling languages such as UML or OML, and implementation languages such as Java, SQL, or CORBA-IDL.
- *Producers:* active entities (human or nonhuman) that develop the work products.
- *Work Units:* operations that are performed by producers when developing work products. One or more *producers* develop a *work product* during the execution of one or more *work units*.
- *Stages:* durations or points in time that provide a high-level organization to the work units.

OPF: Work Units

- ***Activity:***
  - a major work unit consisting of a related collection of jobs that produce a set of work products
  - Coarse-grained descriptions of what needs to be done
  - Some important instances defined by OPEN are: Project Initiation, Requirements Engineering, Analysis and Model Refinement, Project Planning, and Build
- ***Task:***
  - Smallest atomic unit of work
  - Small-scale jobs associated with and comprising the activities
  - Resulting in the creation, modification, or evaluation of one or more work products
- ***Technique:***
  - Define how the jobs are to be done
  - Ways of doing the tasks and activities

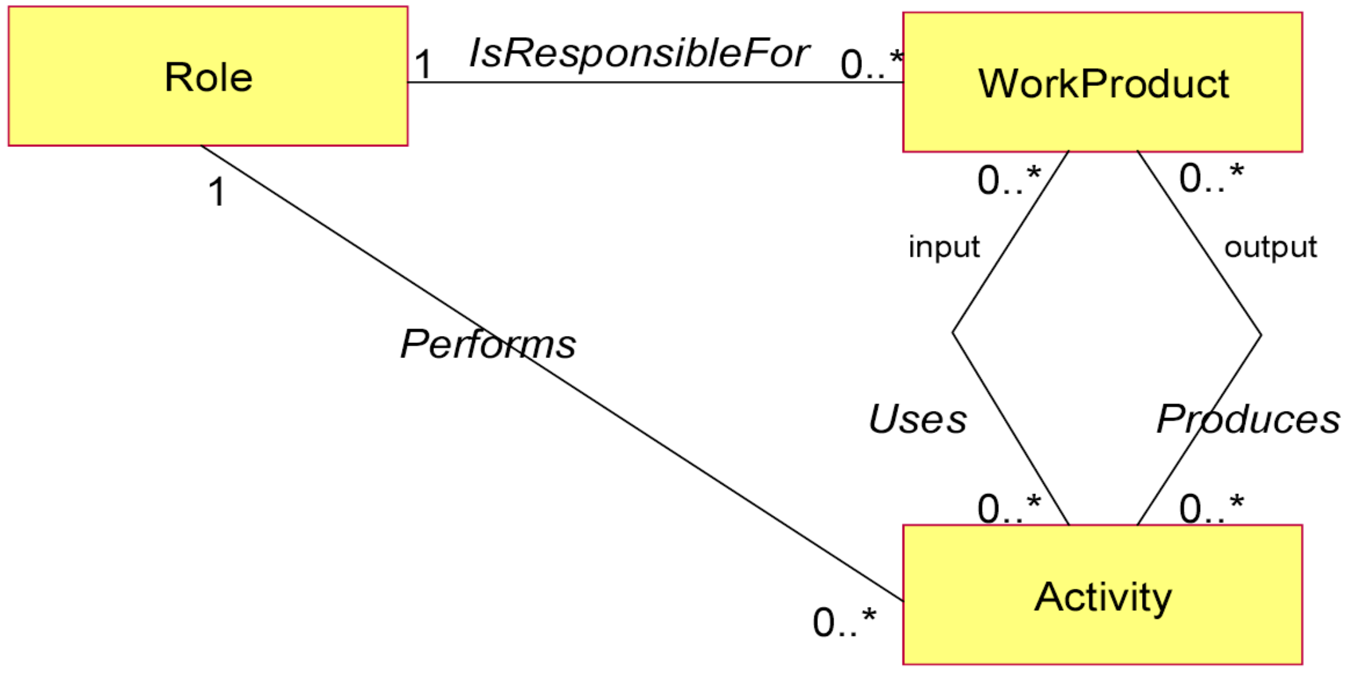
Software Process Engineering Metamodel (SPEM 1.0)

- Similar in essence to OPF yet much simpler
- Primarily based on Rational Corporation's *Unified Software Process Metamodel (USPM)*, which was chiefly intended as a metamodel for the RUP process
- Mainly supports the modeling of UML-based processes similar to RUP
- Unlike OPF, SPEM 1.0
  - does not include a process component library.
  - does not offer a specific procedure for instantiating a software development process using the metamodel (only well-formedness rules are provided).

SPEM 1.0: Core Structure

- Regards the core structure of a software development process as consisting of:
  - *process roles*
  - *work products*
  - *activities*
- Regards a software development process as
  - a collaboration of active entities (*process roles*)
    - aimed at performing specific operations (*activities*)
      - performed on a set of tangible artefacts (*work products*)
      - continued until the artefacts are brought to a well-defined state, and declared as complete.

SPEM 1.0: Core Structure



[OMG 2002]

SPEM 1.0: Detailed Structure

- *Work products:*
  - may be composed of other work products;
  - can be associated with a state machine.
- *Activities:*
  - can be partitioned into *disciplines* based on their common structural and functional themes;
  - may consist of atomic sub-activities called *steps*;
  - can have a *precondition* and a *goal* as constraints on its enactment;
  - may be associated with an *activity graph,* which shows the flow of steps in the activity.

SPEM 1.0: Lifecycle Definition

- SPEM incorporates definitions for
  - *Iteration*
  - *Phase*
  - *Lifecycle*
- Intended to constrain the order in which the activities are performed, and to define the lifecycle structure of the process
- Very similar to their corresponding definitions in RUP

Software and Systems Process Engineering Metamodel (SPEM 2.0)

- Adopted by OMG in December 2006, and revised in 2008
- Addresses the weaknesses of SPEM 1.0
- Provides necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes
  - Provides standardized representation and managed libraries of reusable method content
  - Supports systematic development, management, and growth of development processes
  - Supports deployment of method content and process needed by defining configurations of processes and method content
  - Supports enactment of process for development projects

SPEM 2.0: Conceptual Usage Framework

Standardize representation and manage libraries of reusable **Method Content**

Content on agile development

Content on managing iterative development

Guidance on serialized java beans

JUnit user guidance

Content on J2EE

Configuration mgmt guidelines

Develop and manage **Processes** for performing projects

Lessons learnt from previous project and iteration

Corporate guidelines on compliance

Process assets patterns

Standard or reference processes

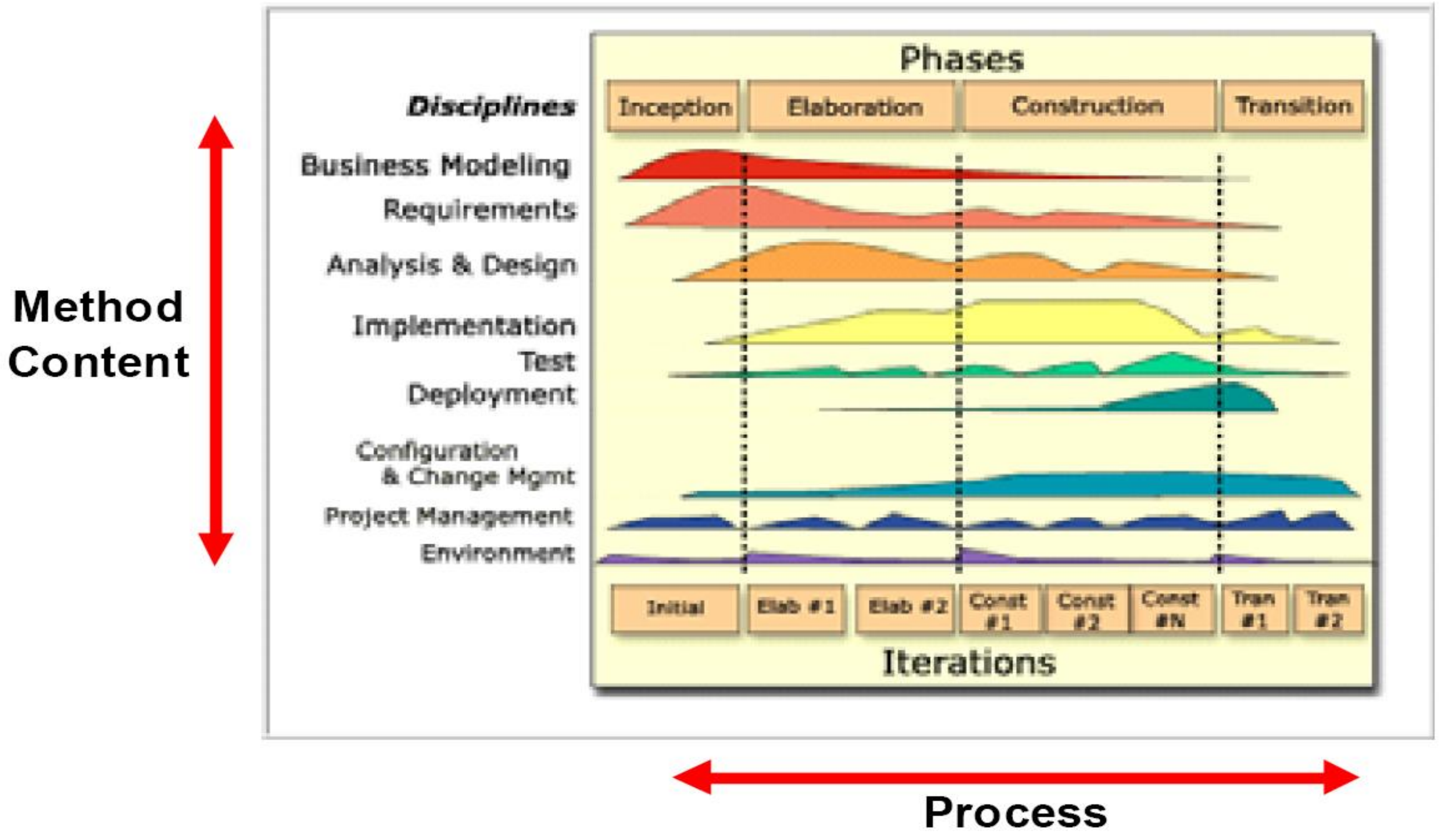
Project plan templates

Configure a cohesive process framework customized for my project needs

Create project plan templates for **Enactment** of process in the context of my project

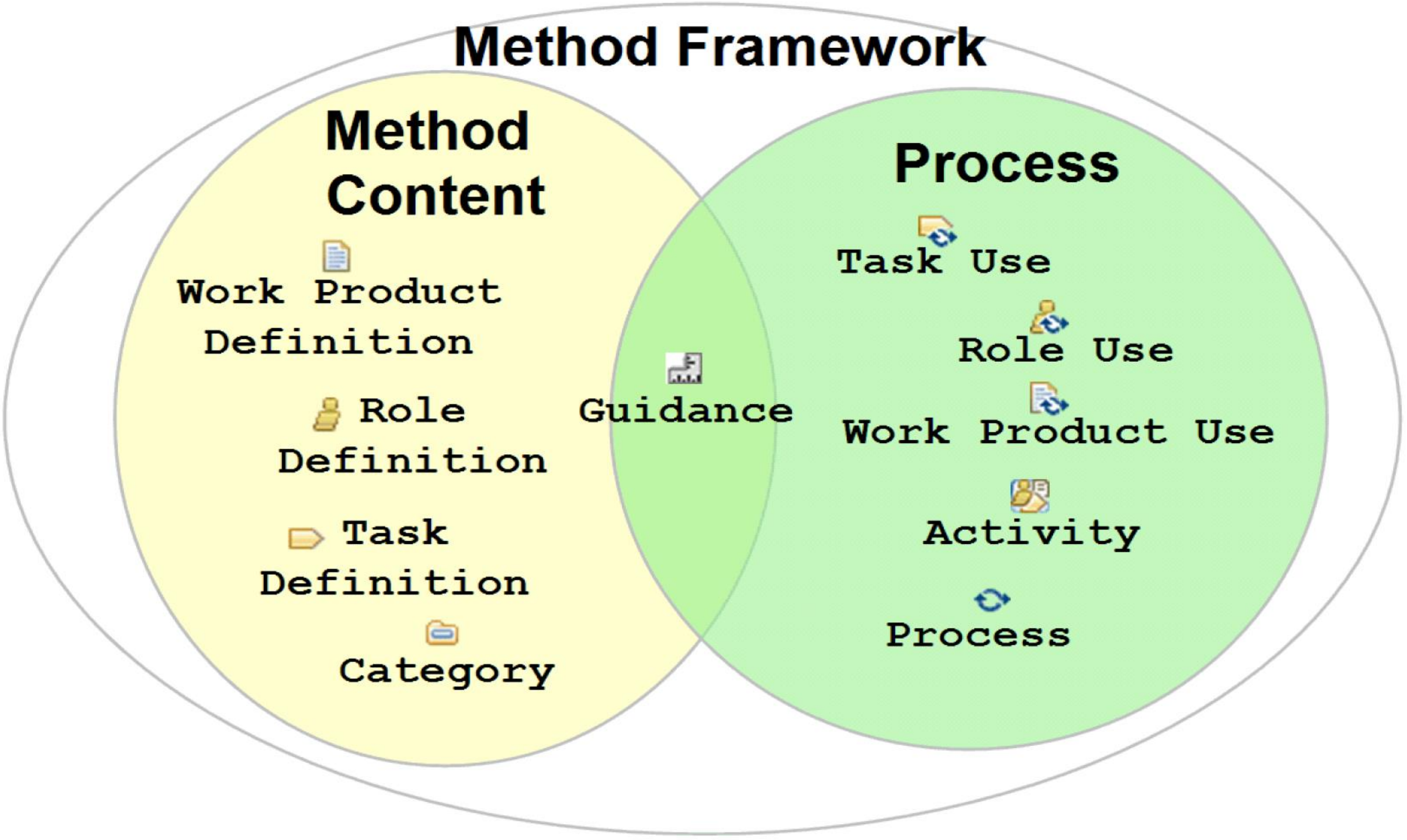
[OMG 2008]

SPEM 2.0: Separation of Method Content from Development Process (1)



[OMG 2008]

SPEM 2.0: Separation of Method Content from Development Process (2)



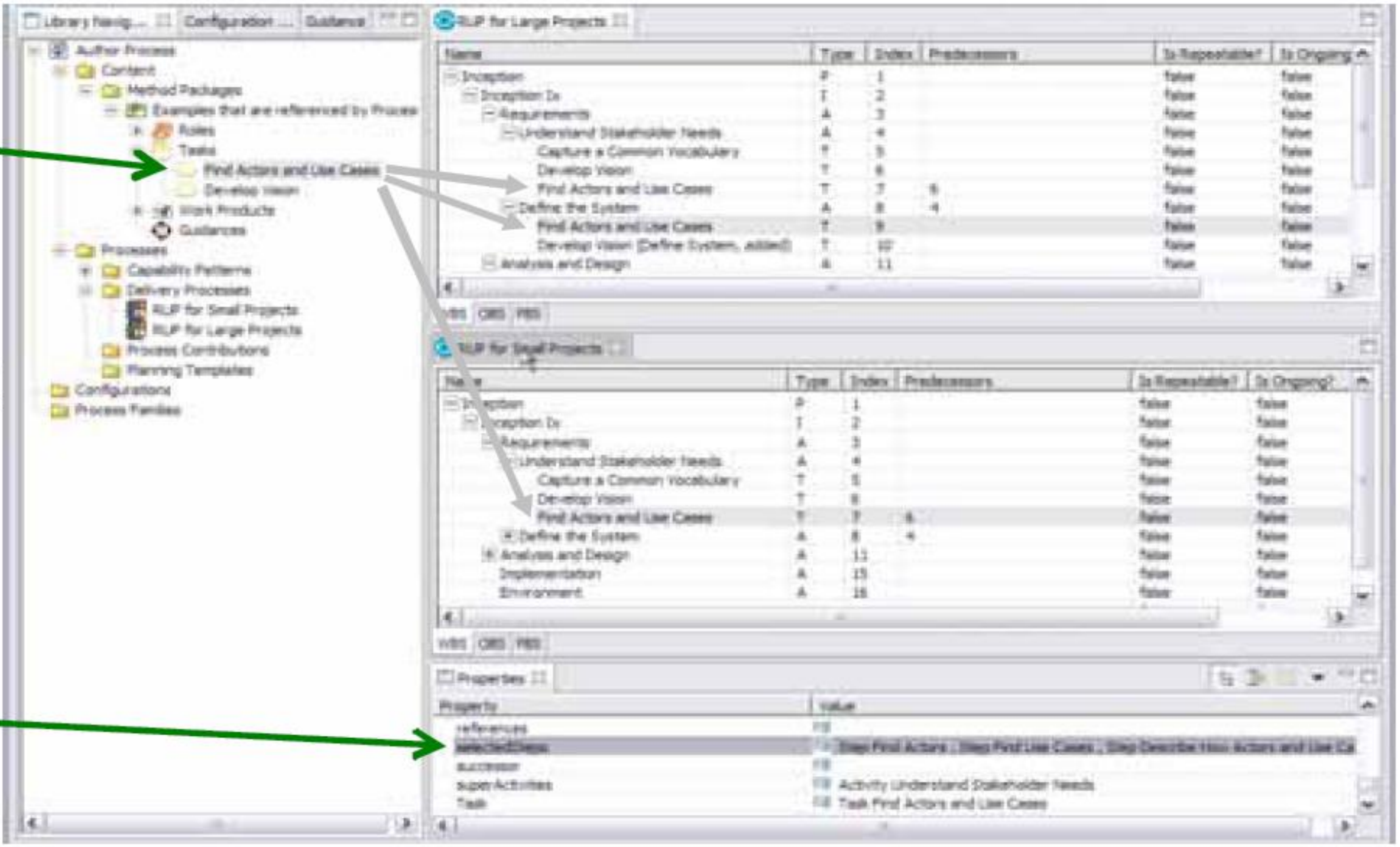
[OMG 2008]

SPEM 2.0: Separation of Method Content from Development Process (3)

Method Content Element "Task Definition" referenced in more then one Process.

Underlying technical concept to support reuse and smart customization: "Task Use"

Individual customization of a "Task Use" by selecting steps, providing additional documentation, etc.



RUP for Large Projects

Name	Type	Index	Predecessors	Is Repeatable?	Is Ongoing?
Inception	P	1		false	false
Inception Ix	T	2		false	false
Requirements	A	3		false	false
Understand Stakeholder Needs	A	4		false	false
Capture a Common Vocabulary	T	5		false	false
Develop Vision	T	6		false	false
Find Actors and Use Cases	T	7	6	false	false
Define the System	A	8	4	false	false
Find Actors and Use Cases	T	9		false	false
Develop Vision (Define System, added)	T	10		false	false
Analysis and Design	A	11		false	false

RUP for Small Projects

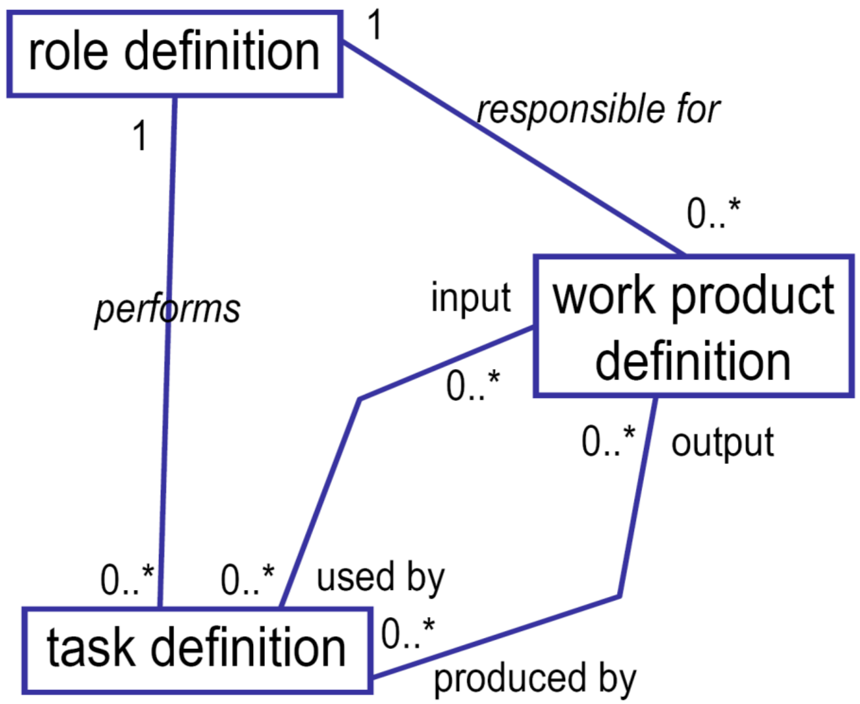
Name	Type	Index	Predecessors	Is Repeatable?	Is Ongoing?
Inception	P	1		false	false
Inception Ix	T	2		false	false
Requirements	A	3		false	false
Understand Stakeholder Needs	A	4		false	false
Capture a Common Vocabulary	T	5		false	false
Develop Vision	T	6		false	false
Find Actors and Use Cases	T	7	6	false	false
Define the System	A	8	4	false	false
Analysis and Design	A	11		false	false
Implementation	A	15		false	false
Environment	A	16		false	false

Properties

Property	Value
references	
selectedSteps	Step Find Actors , Step Find Use Cases , Step Describe How Actors and Use Ca
successor	
superActivities	Activity Understand Stakeholder Needs
Task	Task Find Actors and Use Cases

[OMG 2008]

SPEM 2.0: Method Content - Elements

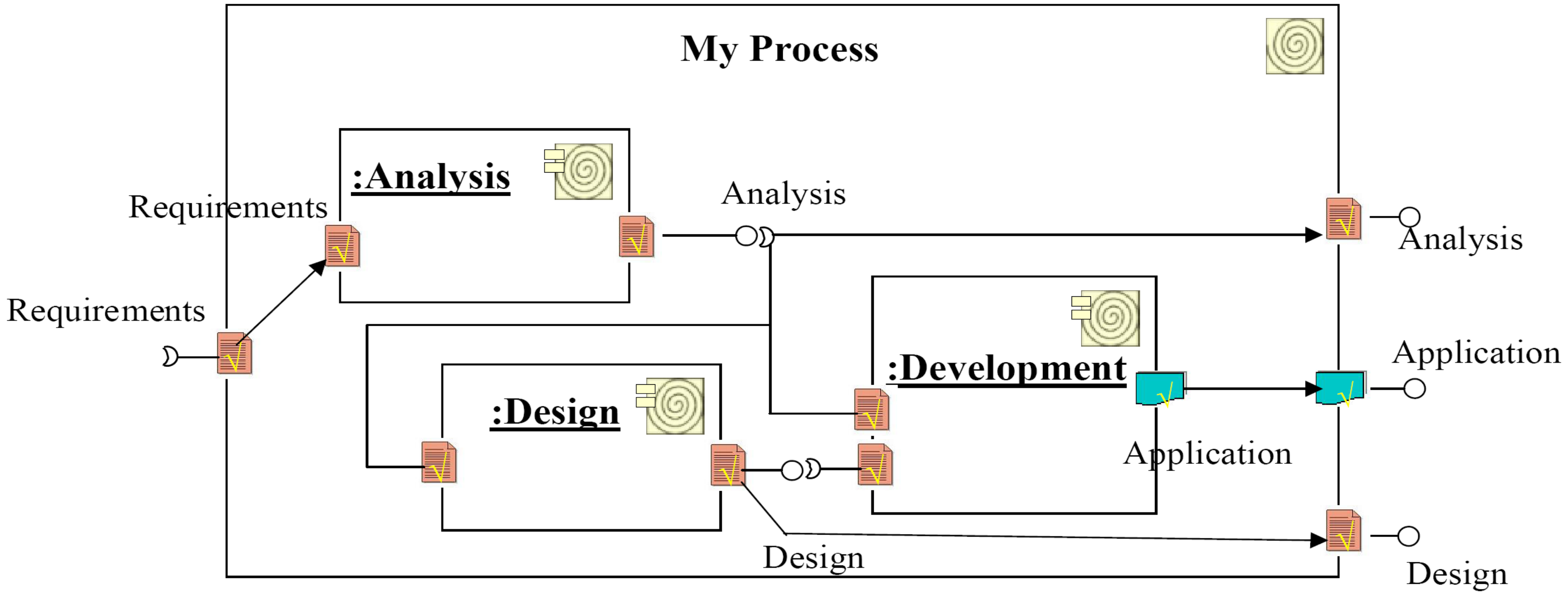


- Roles are responsible for work products
  - Each work product is the responsibility of a single role
- Process roles perform tasks
  - Each task is only performed by a single role
- Work products used as inputs to tasks and outputs from tasks
- “Somebody does something that changes something”

SPEM 2.0: Method Content - Guidance

- Can be associated with any process model element to provide more detailed information about the element to the practitioner
- Can standalone – does not have to be associated
- Most often associated with activities and work products
- SPEM comes with a set of built-in guidance types:
  - Checklist
  - Template
  - Example
  - Tool mentor
  - Guideline

SPEM 2.0: Process Components



- Allow the user to treat the actual definition of the work that produces the outputs as a “black box.”
- Allow different styles or techniques of doing work to be replaced with others.

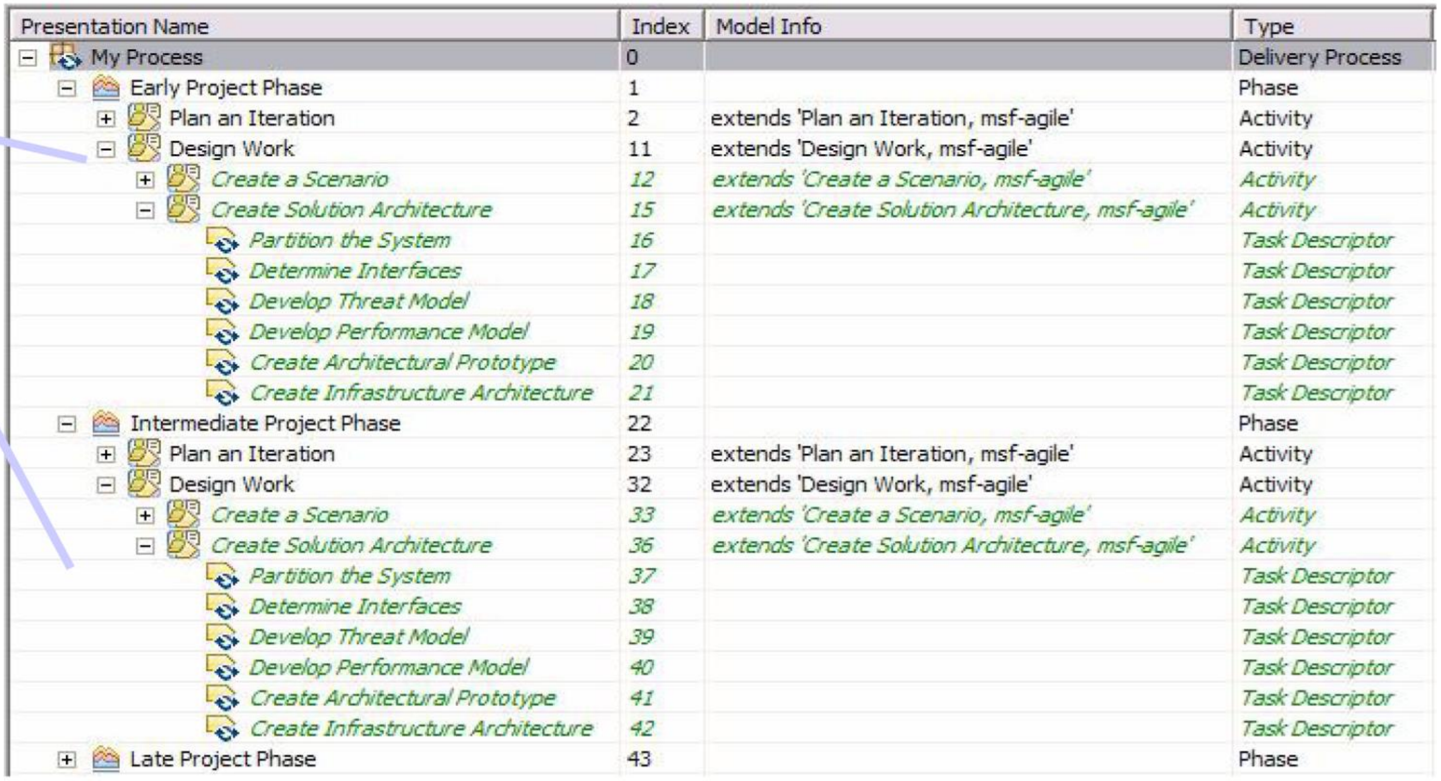
[OMG 2008]

SPEM 2.0: Process Patterns

Revisiting design work based on same underlying pattern

Dynamic linking of patterns increases maintainability

Changes in patterns require zero updates

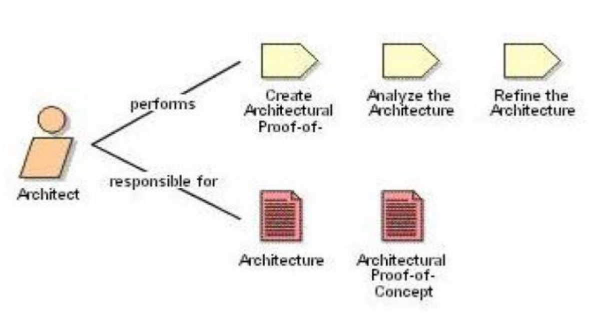


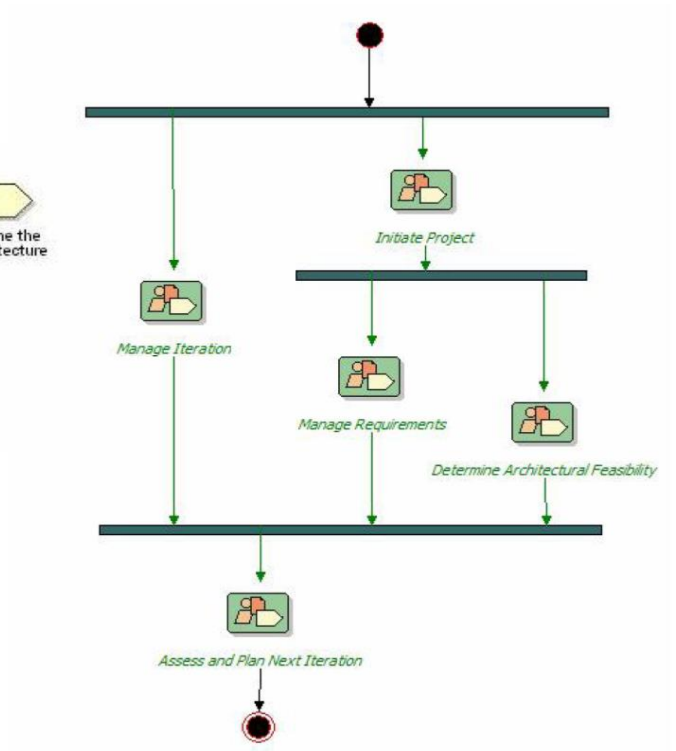
Presentation Name	Index	Model Info	Type
My Process	0		Delivery Process
Early Project Phase	1		Phase
Plan an Iteration	2	extends 'Plan an Iteration, msf-agile'	Activity
Design Work	11	extends 'Design Work, msf-agile'	Activity
Create a Scenario	*12*	*extends 'Create a Scenario, msf-agile'*	*Activity*
Create Solution Architecture	*15*	*extends 'Create Solution Architecture, msf-agile'*	*Activity*
Partition the System	*16*		*Task Descriptor*
Determine Interfaces	*17*		*Task Descriptor*
Develop Threat Model	*18*		*Task Descriptor*
Develop Performance Model	*19*		*Task Descriptor*
Create Architectural Prototype	*20*		*Task Descriptor*
Create Infrastructure Architecture	*21*		*Task Descriptor*
Intermediate Project Phase	22		Phase
Plan an Iteration	23	extends 'Plan an Iteration, msf-agile'	Activity
Design Work	32	extends 'Design Work, msf-agile'	Activity
Create a Scenario	*33*	*extends 'Create a Scenario, msf-agile'*	*Activity*
Create Solution Architecture	*36*	*extends 'Create Solution Architecture, msf-agile'*	*Activity*
Partition the System	*37*		*Task Descriptor*
Determine Interfaces	*38*		*Task Descriptor*
Develop Threat Model	*39*		*Task Descriptor*
Develop Performance Model	*40*		*Task Descriptor*
Create Architectural Prototype	*41*		*Task Descriptor*
Create Infrastructure Architecture	*42*		*Task Descriptor*
Late Project Phase	43		Phase

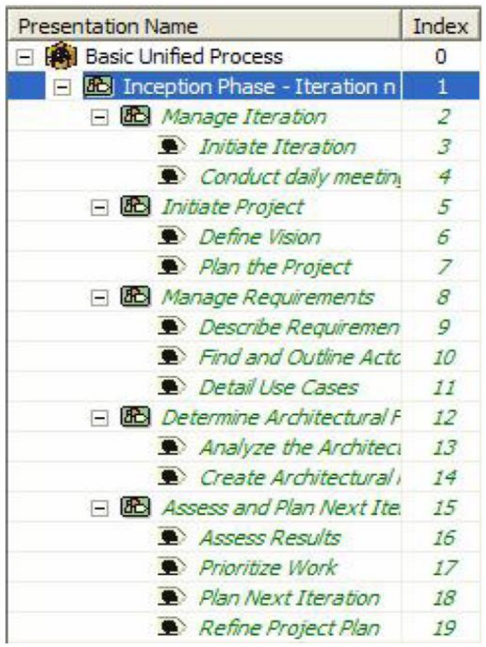
[OMG 2008]

SPEM 2.0: Modeling Enactable Processes

Role, Tasks, and Work Products	Activity Diagram	Work Breakdown Structures




Initiate Project
Manage Iteration
Manage Requirements
Determine Architectural Feasibility
Assess and Plan Next Iteration



Presentation Name	Index
Basic Unified Process	0
Inception Phase - Iteration n	1
Manage Iteration	2
Initiate Iteration	3
Conduct daily meetin	4
Initiate Project	5
Define Vision	6
Plan the Project	7
Manage Requirements	8
Describe Requiremen	9
Find and Outline Actc	10
Detail Use Cases	11
Determine Architectural F	12
Analyze the Architect	13
Create Architectural	14
Assess and Plan Next Ite	15
Assess Results	16
Prioritize Work	17
Plan Next Iteration	18
Refine Project Plan	19

[OMG 2008]

References

- Firesmith, D., Henderson-Sellers, B., *The OPEN Process Framework: An Introduction*. Addison-Wesley, 2001.
- OMG, *Software Process Engineering Metamodel Specification (v1.0).* Object Management Group (OMG), 2002.
- OMG, *Software and Systems Process Engineering Metamodel Specification (v2.0).* Object Management Group (OMG), 2008.